AUTONOMY ARCHITECHTURES FOR A CONSTELLATION OF SPACECRAFT

Anthony Barrett

Jet Propulsion Laboratory, California Institute of Technology

Until the past few years, missions typically involved fairly large expensive spacecraft. Such missions have primarily favored using older proven technologies over more recently developed ones, and humans controlled spacecraft by manually generating detailed command sequences with low-level tools and then transmitting the sequences for subsequent execution on a spacecraft controller.

This approach toward controlling a spacecraft has worked spectacularly on previous missions, but it has limitations deriving from communications restrictions - scheduling time to communicate with a particular spacecraft involves competing with other projects due to the limited number of deep space network antennae. This implies that a spacecraft can spend a long time just waiting whenever a command sequence fails. This is one reason why the New Millennium program has an objective to migrate parts of mission control tasks onboard a spacecraft to reduce wait time by making spacecraft more robust. The migrated software is called a "remote agent" and has 4 components: a mission manager to generate the high level goals, a planner/scheduler to turn goals into activities while reasoning about future expected situations, an executive/diagnostics engine to initiate and maintain activities while interpreting sensed events by reasoning about past and present situations, and a conventional real-time subsystem to interface with the spacecraft to implement an activity's primitive actions.

scheduling

In addition to needing remote planning and execution for isolated spacecraft, a trend toward multiple-spacecraft missions points to the need for remote distributed planning and execution. The past few years have seen missions with growing numbers of probes. Pathfinder has its rover (Sojourner), Cassini has its lander (Huygens), and the New Millenium Deep Space 3 (DS3) proposal involves a constellation of 3 spacecraft for interferometric mapping. This trend is expected to continue to progressively larger fleets. For example, one mission proposed to succeed DS3 would have 18 spacecraft flying in formation in order to detect earth-sized planets orbiting other stars. A proposed magnetospheric constellation would involve 5 to 500 spacecraft in Earth orbit to measure global phenomena within the magnetosphere.

This work describes and compares three autonomy architectures for a system that continuously plans to control a fleet of spacecraft using collective mission goals instead of goals or command sequences for each spacecraft. A fleet of self-commanding spacecraft would autonomously coordinate itself to satisfy high level science and engineering goals in a changing partially-understood environment - making feasible the operation of tens or even a hundred spacecraft (such as for interferometry or plasma physics missions).

The easiest way to adapt autonomous spacecraft research to controlling constellations involves treating the constellation as a single spacecraft. Here one spacecraft directly controls the others as if they were connected. The controlling "master" spacecraft performs all autonomy reasoning, and the slaves only have real-time subsystems to execute the master's commands and transmit local telemetry/observations. The executive/diagnostics module starts actions and the master's real-time subsystem controls the action either locally or remotely through a slave.

While the master/slave approach benefits from conceptual simplicity, it relies on an assumption that the master spacecraft's executive can continuously monitor the slaves' real-time subsystems, and this relies on high-bandwidth highly-reliable communications. Since unintended results occur fairly rarely, one way to relax the bandwidth requirements involves only monitoring unexpected events in spacecraft. Unfortunately, this disables the ability to monitor for unexpected events between spacecraft and leads to a host of coordination problems among the slaves. Also, failures in the communications system can result in losing slaves.

The other two architectures improve robustness while reducing communications by progressively distributing more of the other three remote agent components across the constellation. In a teamwork architecture, all spacecraft have executives and real-time subsystems - only the leader has the planner/scheduler and mission manager. Finally, distributing all remote agent components leads to a peer-to-peer approach toward constellation control.